

NAME

rpict - generate a RADIANCE picture

SYNOPSIS

rpict [**options**] [**\$EVAR**] [**@file**] [**octree**]
rpict [**options**] **-defaults**

DESCRIPTION

Rpict generates a picture from the RADIANCE scene given in *octree* and sends it to the standard output. If no *octree* is given, the standard input is read. (The octree may also be specified as the output of a command enclosed in quotes and preceded by a '!'.) Options specify the viewing parameters as well as giving some control over the calculation. Options may be given on the command line and/or read from the environment and/or read from a file. A command argument beginning with a dollar sign ('\$') is immediately replaced by the contents of the given environment variable. A command argument beginning with an at sign('@') is immediately replaced by the contents of the given file.

In the second form shown above, the default values for the options (modified by those options present) are printed with a brief explanation.

Most options are followed by one or more arguments, which must be separated from the option and each other by white space. The exceptions to this rule are the *-vt* option and the boolean options. Normally, the appearance of a boolean option causes a feature to be "toggled", that is switched from off to on or on to off depending on its previous state. Boolean options may also be set explicitly by following them immediately with a '+' or '-', meaning on or off, respectively. Synonyms for '+' are any of the characters "yYtT1", and synonyms for '-' are any of the characters "nNfF0". All other characters will generate an error.

- vt** Set view type to *t*. If *t* is 'v', a perspective view is selected. If *t* is 'l', a parallel view is used. A cylindrical panorama may be selected by setting *t* to the letter 'c'. This view is like a standard perspective vertically, but projected on a cylinder horizontally (like a soupcan's-eye view). Three fisheye views are provided as well; 'h' yields a hemispherical fisheye view, 'a' results in angular fisheye distortion, and 's' results in a planisphere (stereographic) projection. A hemispherical fisheye is a projection of the hemisphere onto a circle. The maximum view angle for this type is 180 degrees. An angular fisheye view is defined such that distance from the center of the image is proportional to the angle from the central view direction. An angular fisheye can display a full 360 degrees. A planisphere fisheye view maintains angular relationships between lines, and is commonly used for sun path analysis. This is more commonly known as a "stereographic projection," but we avoid the term here so as not to confuse it with a stereoscopic pair. A planisphere fisheye can display up to (but not including) 360 degrees, although distortion becomes extreme as this limit is approached. Note that there is no space between the view type option and its single letter argument.
- vp** *x y z* Set the view point to *x y z*. This is the focal point of a perspective view or the center of a parallel projection.
- vd** *xd yd zd* Set the view direction vector to *xd yd zd*. The length of this vector indicates the focal distance as needed by the *-pd* option, described below.
- vu** *xd yd zd* Set the view up vector (vertical direction) to *xd yd zd*.
- vh** *val* Set the view horizontal size to *val*. For a perspective projection (including fisheye views), *val* is the horizontal field of view (in degrees). For a parallel projection, *val* is the view width in world coordinates.
- vv** *val* Set the view vertical size to *val*.
- vo** *val* Set the view fore clipping plane at a distance of *val* from the view point. The plane will be perpendicular to the view direction for perspective and parallel view types. For fisheye view types, the clipping plane is actually a clipping sphere, centered on the view point with radius *val*. Objects in front of this imaginary surface will not be visible. This may be useful for

seeing through walls (to get a longer perspective from an exterior view point) or for incremental rendering. A value of zero implies no foreground clipping. A negative value produces some interesting effects, since it creates an inverted image for objects behind the viewpoint. This possibility is provided mostly for the purpose of rendering stereographic holograms.

- va** *val* Set the view aft clipping plane at a distance of *val* from the view point. Like the view fore plane, it will be perpendicular to the view direction for perspective and parallel view types. For fisheye view types, the clipping plane is actually a clipping sphere, centered on the view point with radius *val*. Objects behind this imaginary surface will not be visible. A value of zero means no aft clipping, and is the only way to see infinitely distant objects such as the sky.
- vs** *val* Set the view shift to *val*. This is the amount the actual image will be shifted to the right of the specified view. This option is useful for generating skewed perspectives or rendering an image a piece at a time. A value of 1 means that the rendered image starts just to the right of the normal view. A value of -1 would be to the left. Larger or fractional values are permitted as well.
- vl** *val* Set the view lift to *val*. This is the amount the actual image will be lifted up from the specified view, similar to the -vs option.
- vf** *file* Get view parameters from *file*, which may be a picture or a file created by rvu (with the "view" command).
- x** *res* Set the maximum x resolution to *res*.
- y** *res* Set the maximum y resolution to *res*.
- pa** *rat* Set the pixel aspect ratio (height over width) to *rat*. Either the x or the y resolution will be reduced so that the pixels have this ratio for the specified view. If *rat* is zero, then the x and y resolutions will adhere to the given maxima.
- ps** *size* Set the pixel sample spacing to the integer *size*. This specifies the sample spacing (in pixels) for adaptive subdivision on the image plane.
- pt** *frac* Set the pixel sample tolerance to *frac*. If two samples differ by more than this amount, a third sample is taken between them.
- pj** *frac* Set the pixel sample jitter to *frac*. Distributed ray-tracing performs anti-aliasing by randomly sampling over pixels. A value of one will randomly distribute samples over full pixels. A value of zero samples pixel centers only. A value between zero and one is usually best for low-resolution images.
- pm** *frac* Set the pixel motion blur to *frac*. In an animated sequence, the exact view will be blurred between the previous view and the next view as though a shutter were open this fraction of a frame time. (See the -S option regarding animated sequences.) The first view will be blurred according to the difference between the initial view set on the command line and the first view taken from the standard input. It is not advisable to use this option in combination with the *pmbur(1)* program, since one takes the place of the other. However, it may improve results with *pmbur* to use a very small fraction with the -pm option, to avoid the ghosting effect of too few time samples.
- pd** *dia* Set the pixel depth-of-field aperture to a diameter of *dia* (in world coordinates). This will be used in conjunction with the view focal distance, indicated by the length of the view direction vector given in the -vd option. It is not advisable to use this option in combination with the *pdfblur(1)* program, since one takes the place of the other. However, it may improve results with *pdfblur* to use a very small fraction with the -pd option, to avoid the ghosting effect of too few samples.
- dj** *frac* Set the direct jittering to *frac*. A value of zero samples each source at specific sample points (see the -ds option below), giving a smoother but somewhat less accurate rendering. A positive value causes rays to be distributed over each source sample according to its size, resulting in more accurate penumbras. This option should never be greater than 1, and may even cause

problems (such as speckle) when the value is smaller. A warning about aiming failure will be issued if *frac* is too large. It is usually wise to turn off image sampling when using direct jitter by setting *-ps* to 1.

- ds *frac*** Set the direct sampling ratio to *frac*. A light source will be subdivided until the width of each sample area divided by the distance to the illuminated point is below this ratio. This assures accuracy in regions close to large area sources at a slight computational expense. A value of zero turns source subdivision off, sending at most one shadow ray to each light source.
- dt *frac*** Set the direct threshold to *frac*. Shadow testing will stop when the potential contribution of at least the next and at most all remaining light source samples is less than this fraction of the accumulated value. (See the *-dc* option below.) The remaining light source contributions are approximated statistically. A value of zero means that all light source samples will be tested for shadow.
- dc *frac*** Set the direct certainty to *frac*. A value of one guarantees that the absolute accuracy of the direct calculation will be equal to or better than that given in the *-dt* specification. A value of zero only insures that all shadow lines resulting in a contrast change greater than the *-dt* specification will be calculated.
- dr *N*** Set the number of relays for secondary sources to *N*. A value of 0 means that secondary sources will be ignored. A value of 1 means that sources will be made into first generation secondary sources; a value of 2 means that first generation secondary sources will also be made into second generation secondary sources, and so on.
- dp *D*** Set the secondary source presampling density to *D*. This is the number of samples per steradian that will be used to determine ahead of time whether or not it is worth following shadow rays through all the reflections and/or transmissions associated with a secondary source path. A value of 0 means that the full secondary source path will always be tested for shadows if it is tested at all.
- dv** Boolean switch for light source visibility. With this switch off, sources will be black when viewed directly although they will still participate in the direct calculation. This option may be desirable in conjunction with the *-i* option so that light sources do not appear in the output.
- ss *samp*** Set the specular sampling to *samp*. For values less than 1, this is the degree to which the highlights are sampled for rough specular materials. A value greater than one causes multiple ray samples to be sent to reduce noise at a commensurate cost. A value of zero means that no jittering will take place, and all reflections will appear sharp even when they should be diffuse. This may be desirable when used in combination with image sampling (see *-ps* option above) to obtain faster renderings.
- st *frac*** Set the specular sampling threshold to *frac*. This is the minimum fraction of reflection or transmission, under which no specular sampling is performed. A value of zero means that highlights will always be sampled by tracing reflected or transmitted rays. A value of one means that specular sampling is never used. Highlights from light sources will always be correct, but reflections from other surfaces will be approximated using an ambient value. A sampling threshold between zero and one offers a compromise between image accuracy and rendering time.
- bv** Boolean switch for back face visibility. With this switch off, back faces of all objects will be invisible to view rays. This is dangerous unless the model was constructed such that all surface normals face outward. Although turning off back face visibility does not save much computation time under most circumstances, it may be useful as a tool for scene debugging, or for seeing through one-sided walls from the outside.
- av *red grn blu*** Set the ambient value to a radiance of *red grn blu*. This is the final value used in place of an indirect light calculation. If the number of ambient bounces is one or greater and the ambient value weight is non-zero (see *-aw* and *-ab* below), this value may be modified by the computed

indirect values to improve overall accuracy.

- aw** *N* Set the relative weight of the ambient value given with the *-av* option to *N*. As new indirect irradiances are computed, they will modify the default ambient value in a moving average, with the specified weight assigned to the initial value given on the command and all other weights set to 1. If a value of 0 is given with this option, then the initial ambient value is never modified. This is the safest value for scenes with large differences in indirect contributions, such as when both indoor and outdoor (daylight) areas are visible.
- ab** *N* Set the number of ambient bounces to *N*. This is the maximum number of diffuse bounces computed by the indirect calculation. A value of zero implies no indirect calculation.
- ar** *res* Set the ambient resolution to *res*. This number will determine the maximum density of ambient values used in interpolation. Error will start to increase on surfaces spaced closer than the scene size divided by the ambient resolution. The maximum ambient value density is the scene size times the ambient accuracy (see the *-aa* option below) divided by the ambient resolution. The scene size can be determined using *getinfo(1)* with the *-d* option on the input octree. A value of zero is interpreted as unlimited resolution.
- aa** *acc* Set the ambient accuracy to *acc*. This value will approximately equal the error from indirect illuminance interpolation. A value of zero implies no interpolation.
- ad** *N* Set the number of ambient divisions to *N*. The error in the Monte Carlo calculation of indirect illuminance will be inversely proportional to the square root of this number. A value of zero implies no indirect calculation.
- as** *N* Set the number of ambient super-samples to *N*. Super-samples are applied only to the ambient divisions which show a significant change.
- af** *fname* Set the ambient file to *fname*. This is where indirect illuminance will be stored and retrieved. Normally, indirect illuminance values are kept in memory and lost when the program finishes or dies. By using a file, different invocations can share illuminance values, saving time in the computation. Also, by creating an ambient file during a low resolution rendering, better results can be obtained in a second high resolution pass. The ambient file is in a machine-independent binary format which may be examined with *lookamb(1)*.

The ambient file may also be used as a means of communication and data sharing between simultaneously executing processes. The same file may be used by multiple processes, possibly running on different machines and accessing the file via the network (ie. *nfs(4)*). The network lock manager *lockd(8)* is used to insure that this information is used consistently.

If any calculation parameters are changed or the scene is modified, the old ambient file should be removed so that the calculation can start over from scratch. For convenience, the original ambient parameters are listed in the header of the ambient file. *Getinfo(1)* may be used to print out this information.
- ae** *mod* Append *mod* to the ambient exclude list, so that it will not be considered during the indirect calculation. This is a hack for speeding the indirect computation by ignoring certain objects. Any object having *mod* as its modifier will get the default ambient level rather than a calculated value. Any number of excluded modifiers may be given, but each must appear in a separate option.
- ai** *mod* Add *mod* to the ambient include list, so that it will be considered during the indirect calculation. The program can use either an include list or an exclude list, but not both.
- aE** *file* Same as *-ae*, except read modifiers to be excluded from *file*. The RAYPATH environment variable determines which directories are searched for this file. The modifier names are separated by white space in the file.
- aI** *file* Same as *-ai*, except read modifiers to be included from *file*.

-me *rest gext bext*

Set the global medium extinction coefficient to the indicated color, in units of 1/distance (distance in world coordinates). Light will be scattered or absorbed over distance according to this value. The ratio of scattering to total scattering plus absorption is set by the albedo parameter, described below.

-ma *ralb galb balb*

Set the global medium albedo to the given value between 0 0 0 and 1 1 1. A zero value means that all light not transmitted by the medium is absorbed. A unitary value means that all light not transmitted by the medium is scattered in some new direction. The isotropy of scattering is determined by the Heyney-Greenstein parameter, described below.

-mg *gecc*

Set the medium Heyney-Greenstein eccentricity parameter to *gecc*. This parameter determines how strongly scattering favors the forward direction. A value of 0 indicates perfectly isotropic scattering. As this parameter approaches 1, scattering tends to prefer the forward direction.

-ms *sampdist*

Set the medium sampling distance to *sampdist*, in world coordinate units. During source scattering, this will be the average distance between adjacent samples. A value of 0 means that only one sample will be taken per light source within a given scattering volume.

-i

Boolean switch to compute irradiance rather than radiance values. This only affects the final result, substituting a Lambertian surface and multiplying the radiance by pi. Glass and other transparent surfaces are ignored during this stage. Light sources still appear with their original radiance values, though the *-dv* option (above) may be used to override this.

-u

Boolean switch to control uncorrelated random sampling. When "off", a low-discrepancy sequence is used, which reduces variance but can result in a dithered appearance in specular highlights. When "on", pure Monte Carlo sampling is used in all calculations.

-lr *N*

Limit reflections to a maximum of *N*, if *N* is a positive integer. If *N* is zero, then Russian roulette is used for ray termination, and the *-lw* setting (below) must be positive. If *N* is a negative integer, then this sets the upper limit of reflections past which Russian roulette will be used. In scenes with dielectrics and total internal reflection, a setting of 0 (no limit) may cause a stack overflow.

-lw *frac*

Limit the weight of each ray to a minimum of *frac*. During ray-tracing, a record is kept of the estimated contribution (weight) a ray would have in the image. If this weight is less than the specified minimum and the *-lr* setting (above) is positive, the ray is not traced. Otherwise, Russian roulette is used to continue rays with a probability equal to the ray weight divided by the given *frac*.

-S *seqstart*

Instead of generating a single picture based only on the view parameters given on the command line, this option causes *rpict* to read view options from the standard input and for each line containing a valid view specification, generate a corresponding picture. This option is most useful for generating animated sequences, though it may also be used to control *rpict* from a remote process for network-distributed rendering. *Seqstart* is a positive integer that will be associated with the first output frame, and incremented for successive output frames. By default, each frame is concatenated to the output stream, but it is possible to change this action using the *-o* option (described below). Multiple frames may be later extracted from the output using *ra_rgbe(1)*.

Note that the octree may not be read from the standard input when using this option.

-o *fspec*

Send the picture(s) to the file(s) given by *fspec* instead of the standard output. If this option is used in combination with *-S* and *fspec* contains an integer field for *printf(3)* (eg. "%03d") then the actual output file name will include the current frame number. *Rpict* will not allow a picture file to be clobbered (overwritten) with this option. If an image in a sequence already exists (*-S* option), *rpict* will skip until it reaches an image that doesn't, or the end of the sequence. This is useful for running *rpict* on multiple machines or processors to render the

same sequence, as each process will skip to the next frame that needs rendering.

- r *fn*** Recover pixel information from the file *fn*. If the program gets killed during picture generation, the information may be recovered using this option. The view parameters and picture dimensions are also recovered from *fn* if possible. The other options should be identical to those which created *fn*, or an inconsistent picture may result. If *fn* is identical to the file specification given with the *-o* option, *rpict* will rename the file prior to copying its contents. This insures that the old file is not overwritten accidentally. (See also the *-ro* option, below.)

If *fn* is an integer and the recover option is used in combination with the *-S* option, then *rpict* skips a number of view specifications on its input equal to the difference between *fn* and *seqstart*. *Rpict* then performs a recovery operation on the file constructed from the frame number *fn* and the output file specification given with the *-o* option. This provides a convenient mechanism for recovering in the middle of an aborted picture sequence.

The recovered file will be removed if the operation is successful. If the recover operation fails (due to lack of disk space) and the output file and recover file specifications are the same, then the original information may be left in a renamed temporary file. (See FILES section, below.)
- ro *fspec*** This option causes pixel information to be recovered from and subsequently returned to the picture file *fspec*. The effect is the same as specifying identical recover and output file names with the *-r* and *-o* options.
- z *fspec*** Write pixel distances out to the file *fspec*. The values are written as short floats, one per pixel in scanline order, as required by *pinterp(1)*. Similar to the *-o* option, the actual file name will be constructed using *printf* and the frame number from the *-S* option. If used with the *-r* option, *-z* also recovers information from an aborted rendering.
- P *pfile*** Execute in a persistent mode, using *pfile* as the control file. This option must be used together with *-S*, and is incompatible with the recover option (*-r*). Persistent execution means that after reaching end-of-file on its input, *rpict* will fork a child process that will wait for another *rpict* command with the same *-P* option to attach to it. (Note that since the rest of the command line options will be those of the original invocation, it is not necessary to give any arguments besides *-P* for subsequent calls.) Killing the process is achieved with the *kill(1)* command. (The process ID in the first line of *pfile* may be used to identify the waiting *rpict* process.) This option may be less useful than the *-PP* variation, explained below.
- PP *pfile*** Execute in continuous-forking persistent mode, using *pfile* as the control file. The difference between this option and the *-P* option described above is the creation of multiple duplicate processes to handle any number of attaches. This provides a simple and reliable mechanism of memory sharing on most multiprocessing platforms, since the *fork(2)* system call will share memory on a copy-on-write basis. This option may be used with *rpiece(1)* to efficiently render a single image using multiple processors on the same host.
- t *sec*** Set the time between progress reports to *sec*. A progress report writes the number of rays traced, the percentage completed, and the CPU usage to the standard error. Reports are given either automatically after the specified interval, or when the process receives a continue (CONT) signal (see *kill(1)*). A value of zero turns automatic reporting off.
- e *efile*** Send error messages and progress reports to *efile* instead of the standard error.
- w** Boolean switch for warning messages. The default is to print warnings, so the first appearance of this option turns them off.

EXAMPLE

```
rpict -vp 10 5 3 -vd 1 -.5 0 scene.oct > scene.hdr
```

```
rpict -S 1 -o frame%02d.hdr scene.oct < keyframes.vf
```

ENVIRONMENT

RAYPATH the directories to check for auxiliary files.

FILES

/tmp/rtXXXXXX	common header information for picture sequence
rfXXXXXX	temporary name for recover file

DIAGNOSTICS

If the program terminates from an input related error, the exit status will be 1. A system related error results in an exit status of 2. If the program receives a signal that is caught, it will exit with a status of 3. In each case, an error message will be printed to the standard error, or to the file designated by the *-e* option.

AUTHOR

Greg Ward

SEE ALSO

getinfo(1), lookamb(1), oconv(1), pdfblur(1), pfilt(1), pinterp(1), pmblur(1), printf(3), ra_rgbe(1), rad(1), rtrace(1), rvu(1)